

**NAME**

**realpath** - returns the canonicalized absolute pathname

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <stdlib.h>
```

*char \**

```
realpath(const char * restrict pathname, char * restrict resolved_path);
```

**DESCRIPTION**

The **realpath**() function resolves all symbolic links, extra "/" characters and references to `./` and `../` in *pathname*, and copies the resulting absolute pathname into the memory pointed to by *resolved\_path*.

The *resolved\_path* argument *must* point to a buffer capable of storing at least `PATH_MAX` characters, or be `NULL`.

The **realpath**() function will resolve both absolute and relative paths and return the absolute pathname corresponding to *pathname*. All components of *pathname* must exist when **realpath**() is called, and all but the last component must name either directories or symlinks pointing to the directories.

**RETURN VALUES**

The **realpath**() function returns *resolved\_path* on success. If the function was supplied `NULL` as *resolved\_path*, and operation did not cause errors, the returned value is a null-terminated string in a buffer allocated by a call to **malloc**(3). If an error occurs, **realpath**() returns `NULL`, and if *resolved\_path* is not `NULL`, the array that it points to contains the pathname which caused the problem.

**ERRORS**

The function **realpath**() may fail and set the external variable *errno* for any of the errors specified for the library functions `lstat`(2), `readlink`(2) and `getcwd`(3).

**SEE ALSO**

`getcwd`(3)

**HISTORY**

The **realpath**() function first appeared in 4.4BSD.

**CAVEATS**

This implementation of **realpath**() differs slightly from the Solaris implementation. The 4.4BSD version

always returns absolute pathnames, whereas the Solaris implementation will, under certain circumstances, return a relative *resolved\_path* when given a relative *pathname*.